

Mateusz SMOLIŃSKI

# SYSTOLICZNE WSPOMAGANIE OBLICZEŃ PRZY OPRACOWANIU RYGORYSTYCZNEJ HISTORII SZEREGOWANIA TRANSAKCJI

**STRESZCZENIE** *Historia szeregowania określa kolejność wykonania transakcji zgłoszonych przez systemy biznesowe wykorzystujące współdzielone bazy danych w środowisku rozproszonym. Występowanie konfliktów pomiędzy operacjami transakcji uwzględnionych w rygorystycznej historii szeregowania uniemożliwia równoległe ich wykonanie. W szczególności historia rygorystyczna eliminuje wystąpienie problemu wzajemnej blokady transakcji rozproszonych. Wymagane obliczenia, niezbędne do przygotowania przez opracowanego planistę rygorystycznej historii wykonania transakcji, są wykonywane przez dedykowany układ sprzętowy. Zaprojektowana liniowa tablica systoliczna zapewnia przetwarzanie równoległe i potokowe przy wykonywaniu obliczeń wspomagających opracowanie rygorystycznej historii szeregowania transakcji.*

**Słowa kluczowe:** *tablica systoliczna, przetwarzane transakcji, historia rygorystyczna, integracja systemów informatycznych*

## 1. KONFLIKTY TRANSAKCJI

---

Systemy informatyczne EIS (ang. *Enterprise Information System*) usprawniające przepływ informacji w biznesie wymagają ciągłej komunikacji.

---

**mgr inż. Mateusz SMOLIŃSKI**  
e-mail: msmolinski@zsk.p.lodz.pl

Zakład Sieci Komputerowych Instytutu Informatyki,  
Politechnika Łódzka

PRACE INSTYTUTU ELEKTROTECHNIKI, zeszyt 246, 2010

Jest to wynik automatyzacji i rosnącej złożoności procesów biznesowych, których realizacja niejednokrotnie wymaga rozbudowy obecnie istniejących lub wdrożenia nowych systemów informatycznych. Zakładając separację lokalizacji składowania i przetwarzania informacji dla pojedynczego systemu informacyjnego, powstaje problem związany z zachowaniem spójności danych przechowywanych w współdzielonych bazach. Systemy informatyczne zgłaszają transakcje, w granicach których są wykonywane operacje na danych zawartych w bazie. W wyniku równoległego wykonania innej transakcji zgłoszona przez EIS transakcja może zostać odwołana. W sytuacji, gdy pojedynczy EIS wykonuje operacje na danych przechowywanych w różnych bazach, wszystkie operacje są wykonywane w ramach zgłoszonej transakcji rozproszonej. W szczególności może wystąpić sytuacja wzajemnej blokady transakcji rozproszonych (ang. *deadlock*), zgłoszonych przez różne systemy informatyczne. Istniejące rozwiązania problemu wystąpienia wzajemnej blokady skutkują odwołaniem przynajmniej jednej z zablokowanych transakcji [2, 11].

Jeżeli w celu synchronizacji wykonania transakcji możliwe jest zapewnienie komunikacji pomiędzy systemami informatycznymi, wówczas przeprowadzenie integracji bazującej na wymianie komunikatów pomiędzy systemami informatycznymi zapewnia rozwiązanie przedstawionych problemów. W dalszej części rozważań analizowano przypadek ogólny, w którym występuje brak bezpośredniej wymiany komunikatów pomiędzy EIS oraz brak komunikacji pomiędzy systemami zarządzania bazami danych DBMS (ang. *DataBase Management System*). Separacja może być spowodowana brakiem interfejsów dostępowych czy też możliwości rozbudowy, ograniczeniami technologicznymi lub licencyjnymi. Dla uproszczenia rozważań założono, że każda z baz jest obsługiwana przez indywidualny system zarządzania bazami danych, a systemy informatyczne są systemami OLTP (ang. *On-Line Transaction Processing*) wykonującymi jednocześnie różne operacje w wielu bazach danych.

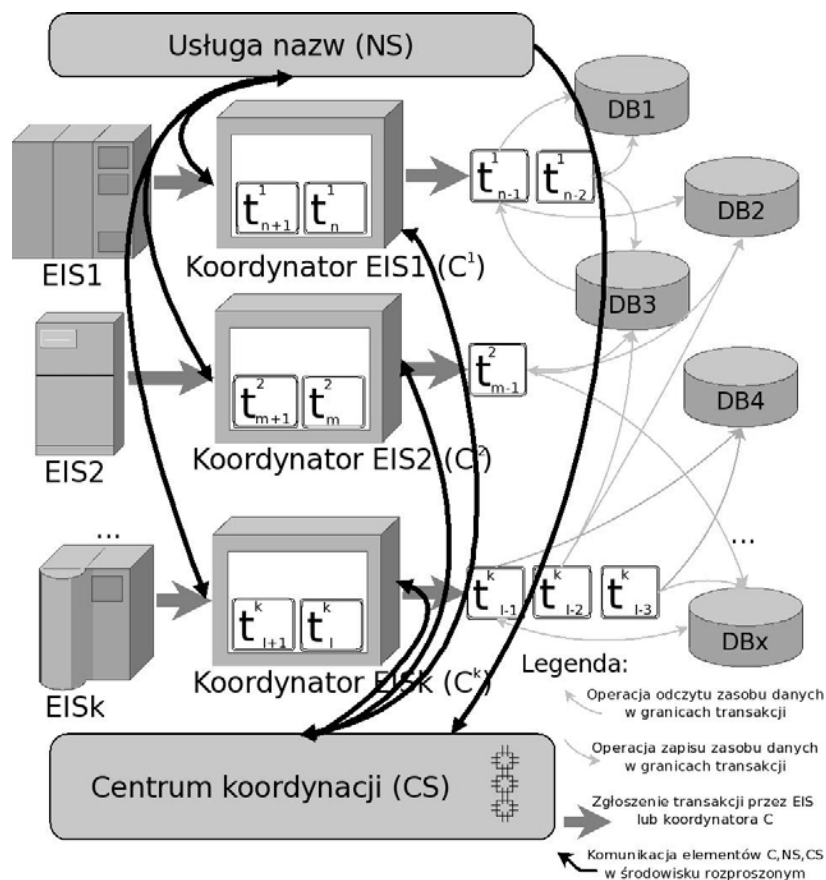
W opisie historii wykonania transakcji posłużono się modelem stron (ang. *page model*) wyróżniającym dwie operacje: odczytu  $r(x)$  i zapisu  $w(x)$  wykonywane na pojedynczym zasobie danych  $x$ . Historia szeregowania  $s$  zawiera wszystkie operacje każdej z transakcji wraz z właściwym jej zakończeniem: zatwierdzeniem transakcji  $c$  (ang. *commit*) lub odwołaniem  $a$  (ang. *rollback/abort*). W literaturze [11], przyjmując model stron, zdefiniowano konflikt operacji względem historii, jeżeli należą one do różnych transakcji występujących w historii i są wykonywane na tym samym zasobie danych, a przynajmniej jedna z nich jest operacją zapisu.

Zatem każde odwołanie transakcji będące następstwem równoległego wykonania innej transakcji wynika wprost z występowania operacji konfliktowych przynależących do wykonywanych równolegle transakcji. Przygotowując historię szeregowania eliminującą konflikty transakcji należy, więc określić

kolejność wykonania dla każdej pary transakcji zawierających operacje konfliktowe. Zgodnie z pozycją [11] literatury, jeżeli wszystkie występujące wcześniej w historii transakcje posiadające operacje konfliktowe zostały zakończone, wówczas historię nazywamy rygorystyczną.

## 2. IDENTYFIKACJA ZASOBÓW TRANSAKCJI

W celu wprowadzenia zcentralizowanego planisty, przygotowującego rygorystyczną historię szeregowania transakcji rozproszonych, zgłaszanych przez EIS wprowadzono w środowisku rozproszonym następujące elementy: koordynatora C (ang. *Coordinator*) dla każdego EIS, usługę nazw NS (ang. *Naming Service*) oraz centrum koordynacji CS (ang. *Coordination Service*) [7, 8, 9, 10].



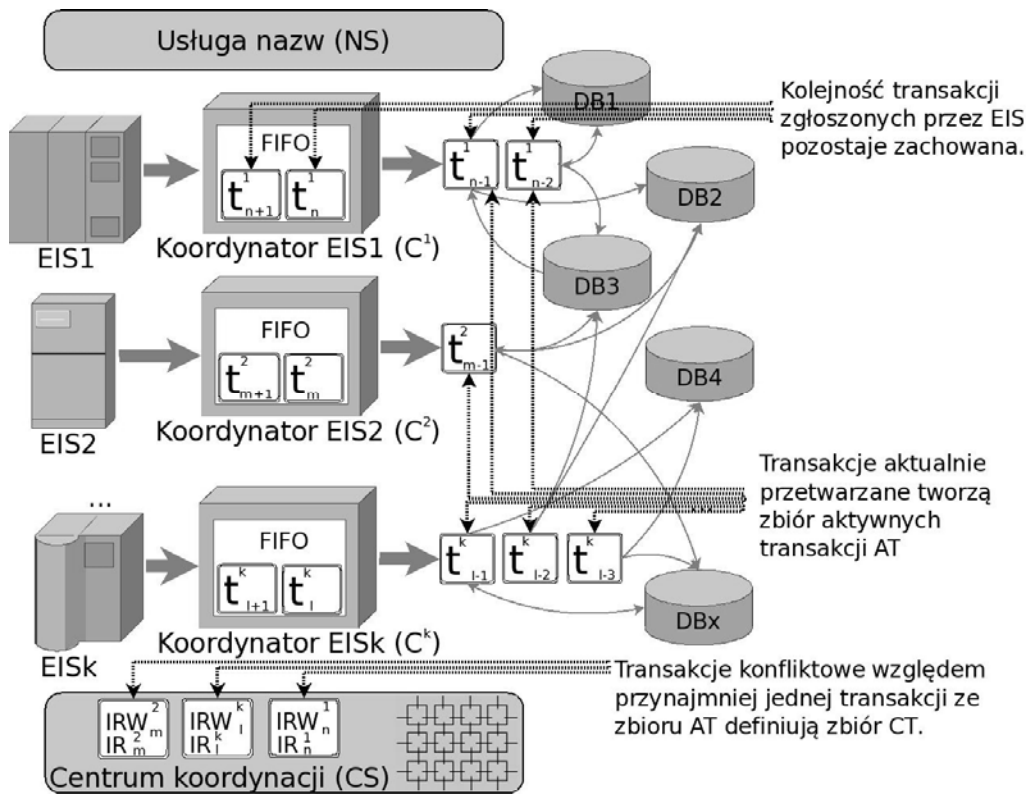
Rys. 1. Koordinatory EIS, usługa nazw NS i centrum koordynacji CS w środowisku rozproszonym

Koordinator jest indywidualnie przypisany do EIS pełniąc funkcję pośrednika, który identyfikuje zgłoszone przez system informatyczny transakcje i wstrzymuje ich wykonanie z zachowaniem kolejności zgłoszeń. Identyfikacja wymaga określenia skończonego zbioru zasobów danych odpowiadającego krotkom, na których wykonywane są operacje w granicach zgłoszonej transakcji. Zachowanie kolejności zgłoszeń transakcji jest zapewnione poprzez zastosowanie kolejki FIFO (ang. *First In First Out*). Wykonanie każdej transakcji, która została umieszczona w kolejce FIFO koordynatora, zostaje wstrzymane. Zadaniem koordynatora jest zarejestrowanie najstarszej w kolejce FIFO transakcji w usłudze nazw w celu uzyskania binarnych identyfikatorów zasobów *IRW* oraz *IR*, a następnie zgłoszenie ich do usługi koordynacji. Kiedy koordynator uzyska żeton wydany przez CS, wówczas najstarsza transakcja jest usuwana z kolejki FIFO, a jej wykonanie zostaje przywrócone. Po zakończeniu transakcji koordynator wyrejestrowuje transakcję z usługi nazw NS, która przekazuje tę informację do CS.

Usługa nazw jest centralnym elementem środowiska rozproszonego, zarządzającym reprezentacją ciągu binarnego o ustalonej długości. Pojedynczy bit zarządzanego ciągu jedynie na określony czas jednoznacznie reprezentuje zasób danych przyporządkowany przez NS. Dla każdego zasobu danych, na którym wykonywane są operacje w granicach rejestrowanej transakcji, przyporządkowywany zostaje jeden bit. Proces rejestracji wymusza inkrementację licznika powiązanego z reprezentacją pojedynczego zasobu danych. Analogicznie, w procesie wyrejestrowania transakcji każdy z liczników odpowiadających reprezentacji poszczególnych zasobów danych transakcji zostaje poddany dekrementacji. Zerowa wartość licznika oznacza, że przyporządkowanie bitowej reprezentacji do zasobu danych zostaje usunięte, a dany bit może w wyniku kolejnych rejestracji reprezentować inny zasób danych.

Każda zarejestrowana w NS transakcja posiada binarne identyfikatory zawierające w swojej reprezentacji bity przypisane do zasobów danych  $x$ , na których w granicach transakcji wykonano operację: odczytu  $r(x)$  lub zapisu  $w(x)$  – identyfikator *IRW*, tylko odczytu  $r(x)$  (bez  $w(x)$ ) – identyfikator *IR*. Binarna reprezentacja zasobów transakcji jest jednoznaczna, ujednoliciła zapis, redukując rozmiar reprezentacji zasobów danych na potrzeby transmisji oraz upraszcza dalsze przetwarzanie tych informacji.

Za opracowanie rygorystycznej historii wykonania transakcji odpowiada planista zlokalizowany w centralnym elemencie koordynującym CS. Oprócz planisty, CS zawiera także dedykowaną liniową tablicę systoliczną wspomagającą obliczenia niezbędne do opracowania rygorystycznej historii wykonania zgłoszonych do CS transakcji. W tym celu wyróżniono dwie listy: transakcji aktualnie wykonywanych AT (ang. *Active Transactions*) oraz transakcji konfliktowych CT (ang. *Conflicted Transactions*).



Rys. 2. Transakcje aktywne AT oraz transakcje konfliktowe CT

Każda zgłoszona do centrum koordynacji CS transakcja zostaje umieszczona na liście AT, jeżeli każda jej operacja nie jest konfliktowa z żadną operacją wykonywaną w granicach każdej aktualnie wykonywanej transakcji. W innych przypadkach transakcja jest uznawana za konfliktową i umieszczana na liście CT. Należy zauważyć, że liczba transakcji konfliktowych na liście CT nigdy nie przekroczy liczby systemów informatycznych EIS. Wynika to z faktu, że wydanie żetonu przez CS dla koordynatora następuje w sytuacji, gdy zgłoszona transakcja została umieszczona na liście AT lub zakończyła się ostatnia transakcja aktywna, która posiadała konfliktowe operacje z zgłoszoną transakcją umieszczoną na liście CT. Z założenia wszystkie transakcje aktywne nie posiadają operacji wzajemnie konfliktowych. Dlatego wszystkie zasoby danych, na których wykonywane są operacje w granicach wykonywanych transakcji aktywnych można przedstawić w reprezentacji binarnej ARW. Binarne reprezentacje zasobów AR dotyczą jedynie tych zasobów danych, na których transakcje aktywne wykonują jedynie odczyt [10].

$$ARW = \sum_{i \in AT} IRW^i \quad (1)$$

$$AR = \sum_{i \in AT} IR^i \quad (2)$$

W przedstawionych wzorach operator sumy określa operację logicznej alternatywy wykonaną na identyfikatorach binarnych poszczególnych transakcji. Analogicznie, uwzględniając zakończenie  $k$ -tej transakcji aktywnej, otrzymano:

$$ARW_k = \sum_{i \in AT, i \neq k} IRW^i \quad (3)$$

$$AR_k = \sum_{i \in AT, i \neq k} IR^i \quad (4)$$

### 3. WYKRYWANIE KONFLIKTÓW TRANSAKCJI

Binarna reprezentacja zasobów transakcji ułatwia identyfikację wystąpienia konfliktowych operacji. Spełnienie warunku określonego wzorem (5) jest równoznaczne z wystąpieniem konfliktowych operacji należących do różnych, zarejestrowanych w usłudze nazw transakcji [10].

$$(IRW^i \text{ and } IRW^j) \text{ xor } (IR^i \text{ and } IR^j) \neq 0 \quad (5)$$

Ponadto, wykrycie konfliktu pomiędzy zgłoszoną do CS przez koordynatora transakcją a transakcjami aktywnymi AT jest równoznaczne spełnieniu warunku określonego wzorem (6):

$$(IRW^i \text{ and } ARW) \text{ xor } (IR^i \text{ and } AR) \neq 0 \quad (6)$$

Wzór (6) określa warunek konieczny i dostateczny, który spełnia każda transakcja konfliktowa umieszczona na liście CT. Zatem transakcja może być przeniesiona na listę transakcji aktywnych jedynie w sytuacji, gdy warunek (6) nie będzie spełniony. Nastąpi to wówczas, gdy zostanie zakończona ostatnia transakcja aktywna, zawierająca operacje konfliktowe z rozpatrywaną transakcją. Czas wykonania każdej aktywnej transakcji jest różny i zależy m. in. od liczby wykonywanych operacji, stąd nie jest możliwe określenie momentu jej zakończenia. Dlatego wyznaczenie kolejności zakończenia transakcji aktywnych nie jest możliwe.

Jednakże w sytuacji zakończenia  $k$ -tej transakcji aktywnej weryfikację braku konfliktu dla każdej transakcji znajdującej się na liście CT można przeprowadzić poprzez sprawdzenie warunku opisanego wzorem (7), wynikającego z zaprzeczenia warunku (6) z uwzględnieniem wzorów (3) i (4) [10].

$$(IRW^i \text{ and } ARW_k) \text{ xor } (IR^i \text{ and } AR_k) = 0 \quad (7)$$

Warunek określony wzorem (7) jest warunkiem, którego spełnienie oznacza brak dotychczas istniejącego konfliktu i jest równoznaczne z przeniesieniem rozpatrywanej transakcji z listy CT na listę AT. Wówczas następuje wydanie żetonu dla koordynatora, który rozpoczyna wykonanie najstarszej wstrzymanej transakcji. Żeton uprawnia też koordynatora do zgłoszenia kolejnej najstarszej wstrzymanej w kolejce FIFO transakcji do CS.

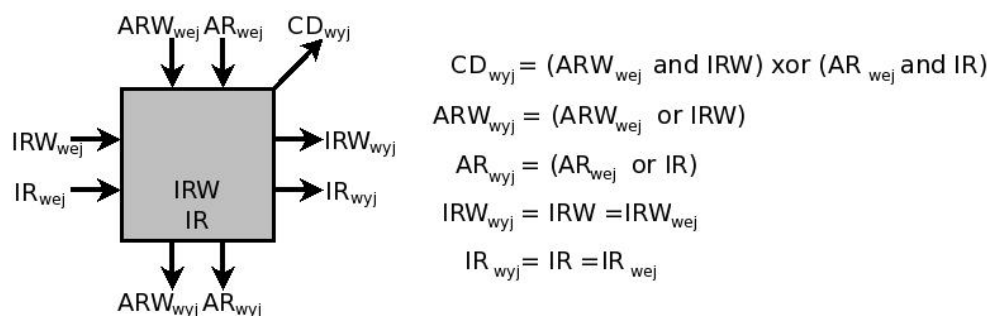
Należy zauważyć, że wykonanie obliczeń niezbędnych do sprawdzenia warunku (7) dla każdej transakcji konfliktowej po zakończeniu  $k$ -tej transakcji aktywnej wprowadza dodatkowe opóźnienia wydłużające czas wstrzymania innych transakcji. Dążąc do wyeliminowania przedstawionych opóźnień należy określić transakcje z listy CT, które mogą być wykonane w sytuacji zakończenia każdej transakcji aktywnej. Przyjmując  $l(AT)$  jako liczbę transakcji aktywnych oraz  $l(CT)$  jako liczbę transakcji konfliktowych, należy sprawdzić  $l(CT)l(AT)$  warunków określonych wzorem (7). Zatem przygotowanie z wyprzedzeniem rygorystycznej historii szeregowania transakcji wymaga wykonania  $l(AT)$ -krotnie więcej obliczeń. W celu wspomaganie obliczeń przy wykrywaniu konfliktów transakcji zaproponowano zastosowanie dedykowanego układu w postaci liniowej tablicy systolicznej. Za wprowadzanie danych oraz interpretację wyników obliczeń systolicznych odpowiada element CS.

W dalszej części rozważań każdy niepusty podzbiór transakcji z listy CT nazywamy scenariuszem, a jego liczność poziomem scenariusza. Dotychczas rozważano przygotowanie dla każdego przypadku zakończenia  $k$ -tej transakcji aktywnej rozszerzenia historii szeregowania o transakcje scenariusza poziomu pierwszego spełniające warunek (7). Eliminacja konfliktów w rozszerzeniu historii o transakcje scenariusza poziomu wyższego oprócz spełnienia warunku (7) wymaga też spełnienia warunku (5) przez transakcje scenariusza. Każdy scenariusz, którego transakcje nie wprowadzają konfliktu operacji do rozszerzenia historii szeregowania nazywamy scenariuszem bezkonfliktowym. Dla każdego przypadku zakończenia  $k$ -tej aktywnej transakcji określenie historii rygorystycznej, zawierającej transakcje najliczniejszego bezkonfliktowego scenariusza, znacząco zwiększa liczbę niezbędnych obliczeń. Zaprojektowana tablica systoliczna wykonuje obliczenia wspomagające wykrywanie konfliktów na potrzeby weryfikacji występowania konfliktów scenariusza dowolnego poziomu [10].

## 4. SYSTOLICZNE WSPOMAGANIE OBLICZEŃ

Zgodnie z definicją tablicy systolicznej przedstawioną w literaturze przez Kunga i Leisersona [3, 4], jest to układ przetwarzający o regularnej modułowej strukturze zbudowany z jednakowych jednostek przetwarzających, połączonych jedynie z najbliższymi jednostkami sąsiednimi, synchronicznie wykonujący elementarne operacje. Wśród cech charakteryzujących tablicę systoliczną należy wymienić: równoległość obliczeń, co umożliwia jednoczesną weryfikację występowania wielu konfliktów transakcji oraz przetwarzanie potokowe binarnych reprezentacji zasobów eliminujące powtarzalne operacje odczytu i zapisu z/do pamięci. Architektura wraz z ściśle dopasowanym algorytmem definiuje tablicę systoliczną. Pulsacją tablicy systolicznej nazywamy jednokrotne wykonanie przez aktywne procesory tablicy w skończonym okresie czasu operacji wymienionych w definicji procesora [5, 6].

Zaprezentowana na rysunku 3 definicja procesora zaprojektowanej tablicy systolicznej określa zbiór wykonywanych operacji oraz kierunki przesyłania poszczególnych danych pomiędzy sąsiednimi procesorami [10].



Rys. 3. Definicja procesora liniowej tablicy systolicznej

Należy zwrócić uwagę, że algorytm systoliczny określa sposób wprowadzania danych do tablicy systolicznej oraz ich przetwarzanie w czasie i przestrzeni (ang. *time-space*), czyli przetwarzanie w kolejnych pulsacjach (ang. *time*) i w odpowiednio rozmieszczonych procesorach (ang. *space*) [6].

Wykonanie operacji przez aktywny procesor w pojedynczej pulsacji tablicy systolicznej jest związane z wczytaniem danych pobranych od sąsiadujących procesorów (tzw. dane wejściowe), wykonaniem obliczeń zgodnie



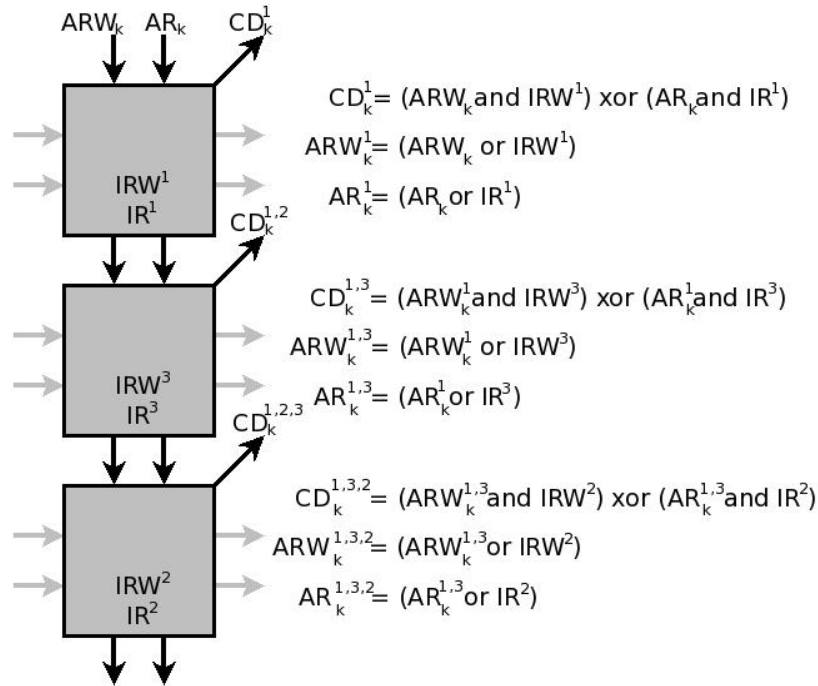
z wzorami przedstawionymi w definicji procesora (uzyskane wyniki obliczeń cząstkowych przechowywane są w rejestrach procesora), wprowadzeniem danych do rejestrów komunikacyjnych celem ich przekazania do sąsiednich procesorów (tzw. dane wyjściowe).

Przed rozpoczęciem obliczeń związanych z wykrywaniem konfliktów transakcji należy przeprowadzić proces wczytania danych dla rozpatrywanego scenariusza do tablicy systolicznej. W procesie wczytywania danych do pojedynczego procesora tablicy systolicznej wprowadzane są binarne identyfikatory zasobów  $IRW$  i  $IR$  pojedynczej transakcji z rozpatrywanego scenariusza. Poziom scenariusza jest ograniczony liczbą systemów EIS występujących w środowisku rozproszonym. W celu zapewnienia możliwości wczytania do tablicy systolicznej scenariusza dowolnego poziomu liczba procesorów liniowej tablicy systolicznej powinna być nie mniejsza niż liczba systemów informatycznych występujących w środowisku rozproszonym. Zatem dla konkretnego środowiska rozproszonego możliwe jest dobranie rozmiaru liniowej tablicy systolicznej [10].

W procesie wczytania scenariusza wykorzystywane są jedynie poziome połączenia pomiędzy procesorami, a w potokowym przetwarzaniu danych przy obliczeniach systolicznych wspomagających wykrywanie konfliktów wyłącznie połączenia pionowe. Binarna reprezentacja zasobów odczytywana przez procesor w pionie  $ARW_{wej}$  oraz  $AR_{wej}$ , zawiera reprezentację wszystkich zasobów danych, na których wykonywane są operacje w rozpatrywanej historii uwzględniającej zakończenie  $k$ -tej transakcji aktywnej. Zgodnie z przedstawioną definicją procesora w obliczeniach systolicznych następuje aktualizacja wczytanych ciągów binarnych i następnie są one przekazywane poprzez pionowe połączenia do kolejnego, sąsiedniego procesora jako  $ARW_{wyj}$  oraz  $AR_{wyj}$ . W aktywnym procesorze liniowej tablicy systolicznej zgodnie z wzorem (7) zostaje też obliczona wartość  $CD_{wyj}$ , na podstawie której można określić występowanie konfliktu w rozszerzonej historii zawierającej scenariusz określonego poziomu. Wartość  $CD_{wyj}=0$  obliczona przez pierwszy procesor liniowej tablicy systolicznej umożliwia stwierdzenie braku konfliktu w rozszerzonej historii zawierającej scenariusz poziomu pierwszego. Drugi procesor obliczając  $CD_{wyj}=0$  pozwala na stwierdzenie braku konfliktu w rozszerzonej historii zawierającej scenariusz poziomu drugiego, jeżeli tylko obliczona wcześniej wartość  $CD_{wyj}$  przez pierwszy procesor też jest zerowa. Ogólnie, jeżeli obliczona przez  $n$  kolejnych procesorów wartość  $CD_{wyj}=0$ , wówczas scenariusz poziomu  $n$  składający się z transakcji, których binarne identyfikatory zostały wczytane do każdego z  $n$  procesorów, jest bezkonfliktowy.

Na rysunku 4 przedstawiono obliczenia systoliczne dla modelu 3-procesorowej tablicy liniowej zbudowanej z procesorów zdefiniowanych na rysunku 3, stosując oznaczenia szczegółowe. Dla uproszczenia przyjęto, że do tablicy wczy-

tano najliczniejszy scenariusz  $S = \{t^1, t^3, t^2\}$  oraz przedstawiono obliczenia zakładające zakończenie tylko jednej  $k$ -tej transakcji aktywnej.



**Rys. 4. Obliczenia systoliczne w modelu 3-procesorowej tablicy liniowej**

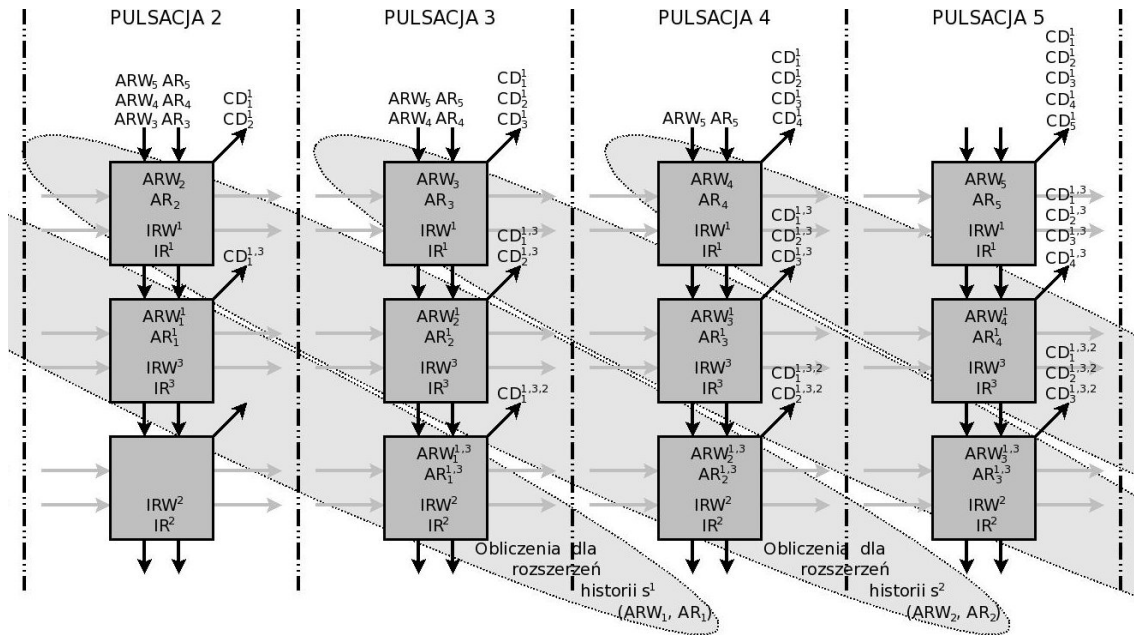
Przyjmując, że  $l(S)$  jest poziomem najliczniejszego scenariusza  $S$  wczytanego do liniowej tablicy systolicznej, wówczas liczba pulsacji  $L(S)$ , w których wykonywane są obliczenia wartości  $CD_{wyj}$ , jest określona wzorem (8).

$$L(S) = l(S) + l(AT) - 1 \quad (8)$$

Liniowa tablica systoliczna w  $L(S)$  pulsacjach zapewnia uzyskanie  $N(S)$  wartości  $CD_{wyj}$ , pozwalających na weryfikację występowania konfliktów w rozpatrywanych rozszerzonych historiach szeregowania transakcji. Wartość  $N(S)$  została określona wzorem (9).

$$N(S) = l(S)l(AT) \quad (9)$$

Na rysunku 5 zaprezentowano przykład obliczeń systolicznych wykonanych w czterech kolejnych pulsacjach modelu 3-procesorowej tablicy liniowej, zakładając wcześniejsze wczytanie scenariusza  $S = \{t^1, t^3, t^2\}$  oraz przyjmując  $l(AT) = 5$ . Zgodnie z wzorem (8) łączna liczba wszystkich pulsacji dla przedstawionych obliczeń systolicznych  $L(S) = 7$ .



Rys. 5. Obliczenia systoliczne w czterech kolejnych pulsacjach dla modelu 3-procesorowej tablicy liniowej, gdzie  $l(AT) = 5$

Liczba wszystkich scenariuszy, dla których wykonywane są obliczenia systoliczne wartości  $CD_{wyj}$  jest uzależniona od liczby transakcji konfliktowych  $l(CT)$ , oraz poziomu wczytywanych scenariuszy. Do określenia liczby możliwych scenariuszy ustalonego poziomu można wykorzystać symbol Newtona. Ze względu na fakt, że tablica systoliczna w pojedynczej pulsacji pozwala na weryfikację występowania konfliktu w historiach rozszerzonych zawierających scenariusze różnych poziomów, część wyników  $CD_{wyj}$ , uzyskanych w obliczeniach systolicznych dla różnych wczytanych scenariuszy może się powtarzać. Tablica systoliczna musi wykonać obliczenia dla każdego z  $M$  unikalnych scenariuszy. Z właściwości symbolu Newtona liczba wszystkich unikalnych scenariuszy  $M$  wyraża się wzorem (10).

$$M = 2^{l(CT)} - 1 \quad (10)$$

## 5. WNIOSKI

Rygorystyczna historia szeregowania transakcji w środowisku rozproszonym pozwala na eliminację konfliktowych operacji dla równoległe wykonywanych transakcji. Brak konfliktów wyklucza występowanie sytuacji wzajem-

nej blokady transakcji oraz zapewnia zachowanie spójności danych biznesowych. Wprowadzenie centralnego planisty wyklucza też sytuacje odwołania transakcji, wynikających jedynie z równoległego wykonywania innych transakcji rozproszonych. Użycie centralnego elementu planującego wymaga zastosowania nowych elementów w środowisku rozproszonym takich jak koordynatorzy przypisani do systemów informatycznych EIS, centralnej usługi nazw NS czy centralnej usługi koordynacji CS. Należy zaznaczyć, że w stosunku do istniejących planistów przygotowujących historie rygorystycznie opracowany planista nie wymaga rozszerzenia zbioru dostępnych w modelu stron operacji wykonywanych na zasobach danych w granicach transakcji [1, 2, 11].

Przedstawiony w artykule planista umożliwia opracowanie z wyprzedzeniem rygorystycznej historii szeregowania zgłoszonych do CS transakcji dla każdego przypadku, uwzględniającego zakończenie pojedynczej transakcji aktywnej. W celu skrócenia czasu obliczeń, zaprojektowano dedykowany układ sprzętowy w postaci liniowej tablicy systolicznej. Podstawowym zdaniem zaprojektowanej tablicy jest wspomaganie wykonania obliczeń niezbędnych przy wykrywaniu konfliktów w rozpatrywanych historiach rozszerzonych. Działanie tablicy systolicznej omówiono na prostym modelu 3-procesorowej tablicy liniowej, prezentując wykonanie obliczeń systolicznych w kolejnych pulsacjach. Dalsze kierunki badań: zwiększenie wydajności obliczeniowej poprzez zastosowanie zestawu połączonych liniowych tablic systolicznych, wykazanie skalowalności opracowanego algorytmu systolicznego oraz zwiększenie niezawodności poprzez wykluczenie pojedynczych punktów występowania awarii w środowisku rozproszonym.

## LITERATURA,

1. Bernstein P. A.: Principles of Transaction Processing, Morgan Kaufmann Publishers, 1997
2. Bernstein P.A., Hadzilacos V., Goodman N.: Concurrency Control and Recovery in Database Systems, Addison-Wesley, 1987.
3. Kung H.T., Leiserson C.E.: Systolic Arrays for (VLSI), Technical Report CMU-CS-79-103, Carnegie-Mellon University, Pittsburg, Pa, 1978
4. Kung S.Y.: VLSI array processors, Prentice-Hall Inc., 1988.
5. Lipowska-Nadolska E., Kwapisz M., Lichy K.: Systoliczne przetwarzanie sygnałów cyfrowych, Akademicka Oficyna Wydawnicza EXIT, 2007.
6. Lipowska-Nadolska E., Morawski M.: Tablice systoliczne. Problemy wybrane., Akademicka Oficyna Wydawnicza PLJ, 1999.
7. Smoliński M: Koncepcja systolicznego akceleratora transakcji rozproszonych., Materiały XV Konferencji Sieci i Systemy Informatyczne – teoria, projekty, wdrożenia, aplikacje. s. 205-206, Łódź, 2007.

8. Smoliński M., Lipowska-Nadolska E.: Koncepcja systolicznego wspomaganie integracji systemów informatycznych., Rozdział 7 w pracy zbiorowej pt. Bazy danych. Rozwój metod i technologii. Architektura, metody formalne i zaawansowana analiza danych, WKiŁ, 2008.
9. Smoliński M., Kwapisz M., Lipowska-Nadolska E.: Scheduling system for distributed transaction processing, Rozdział w monografii: Information Systems Architecture and Technology: Advances in Web-Age Information Systems, Oficyna Wydawnicza Politechniki Wrocławskiej, 2009.
10. Smoliński M.: System szeregowania wykluczający konflikty transakcji rozproszonych z zastosowaniem wspomaganie systolicznego, rozprawa doktorska, 2010.
11. Vossen W.: Transactional Information Systems, Academic Press, 2002.

Rękopis dostarczono dnia 11.05.2010 r.

**Opiniował: dr hab. inż. Stefan F. Filipowicz – prof. PW**

## SYSTOLIC PROCESSING SUPPORT IN DEVELOPING RIGOROUS HISTORY OF TRANSACTION EXECUTION

Mateusz SMOLIŃSKI

**ABSTRACT** *Serializable history defines execution order for every transaction requested by enterprise information systems, that share many databases in distributed environment. When conflict appears between operations, belonging to different transactions from rigorous history, that makes impossible to execute them in parallel. Especially rigorous history eliminates distributed transactions deadlock problem. Required computations, that are necessary to creation of rigorous history, are executed by dedicated hardware unit. Designed linear, systolic array uses parallel and streaming processing when execute computations to create rigorous history.*

**Dr inż. Mateusz SMOLIŃSKI** od ośmiu lat jest pracownikiem naukowo-dydaktycznym Zakładu Sieci Komputerowych Instytutu Informatyki Politechniki Łódzkiej. Zajmuje się zagadnieniami związanymi z projektowaniem, budową, wdrożeniem systemów informatycznych OLTP oraz aspektami dostępności i niezawodności usług sieciowych. Od 5 lat współuczestniczy w przygotowaniu i prowadzeniu szkoleń (kursy oraz studia podyplomowe) związanych z tematyką administracji systemami GNU/Linux realizowanych na Politechnice Łódzkiej.

